

```
In [ ]: #Importing Needed Libraries
```

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import pyodbc #for connecting Ms Sql Server
import warnings
warnings.filterwarnings('ignore')
```

```
In [ ]: #Drivers check
```

```
In [3]: print(pyodbc.dataSources())

{'MS Access Database': 'Microsoft Access Driver (*.mdb, *.accdb)', 'Excel Files': 'Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)', 'MySqlServerDSN': 'SQL Server', 'demo': 'SQL Server', 'demo1': 'SQL Server Native Client 11.0', 'demo2': 'ODBC Driver 17 for SQL Server'}
```

```
In [ ]: #Connecting MS SQL with python
```

```
In [4]: conn = pyodbc.connect(
'Driver=SQL SERVER;'
'Server=LAPTOP-G49RN1PG\SQLEXPRESS;'
'Trusted_Connection=yes;'
'Database=Amazon_Sales;'
)
```

```
In [5]: conn
```

```
Out[5]: <pyodbc.Connection at 0x282712a8370>
```

```
In [ ]: #Below code retrieves and prints a list of table names in the specified database
```

```
In [6]: query = "SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE'"
tables_df = pd.read_sql_query(query, conn)
print(tables_df['TABLE_NAME'].tolist())

['Amazon_Sales']
```

```
In [ ]: #printing head counts to check the database
```

```
In [7]: data = pd.read_sql_query("SELECT * FROM Amazon_Sales", conn)
data.head(6)
```

```
Out[7]:
```

	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_Date	Order_ID	Ship_Date	Units_Sold	Unit_Price	L
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.279999	15
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.699997	11
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.210022	52
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-07-05	8102	9.330000	
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2013-02-06	5062	651.210022	52
5	Australia and Oceania	Solomon Islands	Baby Food	Online	C	2015-02-04	547995746	2015-02-21	2974	255.279999	15

```
In [ ]: #Understanding the data & datatypes
```

```
In [8]: data['Order_Date'] = pd.to_datetime(data['Order_Date'])
data['Ship_Date'] = pd.to_datetime(data['Ship_Date'])
```

```
In [206... data.dtypes
```

```
Out[206... Region          object
Country          object
Item_Type        object
Sales_Channel    object
Order_Priority   object
Order_Date       datetime64[ns]
Order_ID         int64
Ship_Date        datetime64[ns]
Units_Sold       int64
Unit_Price       float64
Unit_Cost        float64
Total_Revenue    float64
Total_Cost       float64
Total_Profit     float64
dtype: object
```

```
In [ ]: 1. What is the total revenue generated by Amazon across all regions and countries?
```

```
In [9]: total_revenue = pd.read_sql_query("SELECT Sum([Total_Revenue]) as Total_Revenue FROM Amazon_Sales", conn)
total_revenue
```

```
Out[9]: Total_Revenue
0      1.373488e+08
```

```
In [ ]: #2. Which country has the highest total profit, and which one has the lowest?
```

```
In [12]: # Print the country with the highest total profit
highest_profit = """
    SELECT [Country], [Total_Profit]
    FROM [Amazon_Sales]
    WHERE [Total_Profit] = (SELECT MAX([Total_Profit]) FROM [Amazon_Sales])
    """
highest_profit = pd.read_sql_query(highest_profit, conn)

# Print the country with the lowest total profit
print("Country with the Highest Total Profit:")
print(highest_profit)

lowest_profit = """
```

```

SELECT [Country], [Total_Profit]
FROM [Amazon_Sales]
WHERE [Total_Profit] = (SELECT MIN([Total_Profit]) FROM [Amazon_Sales])
"""
lowest_profit = pd.read_sql_query(lowest_profit, conn)

# Print the country with the lowest total profit
print("\nCountry with the Lowest Total Profit:")
print(lowest_profit)

```

Country with the Highest Total Profit:

	Country	Total_Profit
0	Pakistan	1719922.0

Country with the Lowest Total Profit:

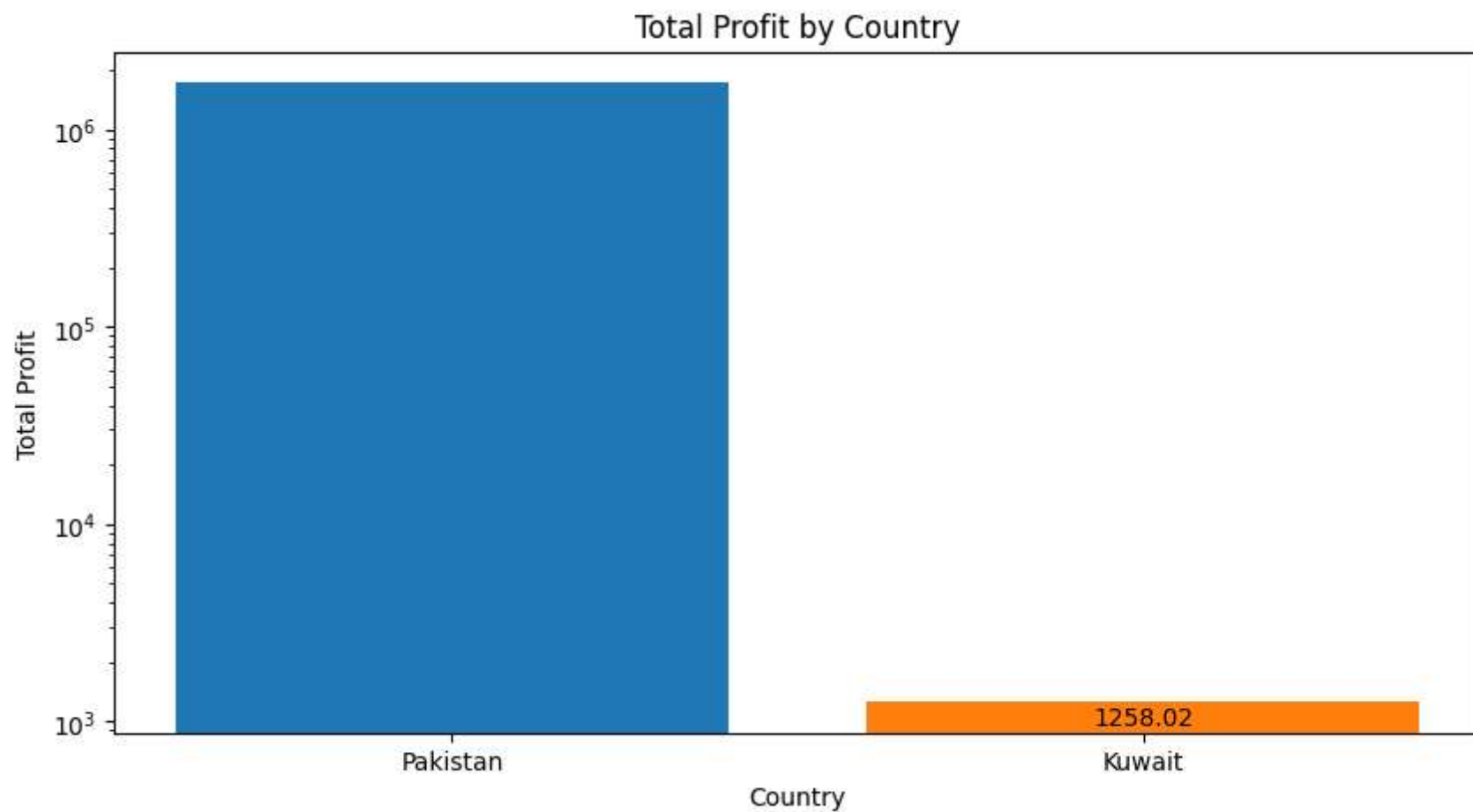
	Country	Total_Profit
0	Kuwait	1258.02002

In []: *#with chart*

```

In [215... plt.figure(figsize = (10,5))
bars = plt.bar(highest_profit["Country"], highest_profit["Total_Profit"])
bars = plt.bar(lowest_profit["Country"], lowest_profit["Total_Profit"])
plt.bar_label(bars, label_type='center', fmt='%.2f')
plt.yscale('log')
plt.xlabel('Country')
plt.ylabel('Total Profit')
plt.title('Total Profit by Country')
plt.show()

```



In []: 3. What is the average unit price and unit cost of items sold in each region?

```
In [13]: query3 = """
        select [Region],
               ROUND(AVG(unit_price), 2) as AVG_UNIT_PRICE,
               ROUND(AVG(unit_cost), 2) AS AVG_UNIT_COST
        From [Amazon_sales]
        group by [Region]
        """

        avg_price_and_cost_of_item = pd.read_sql_query(query3, conn)
        avg_price_and_cost_of_item
```

Out[13]:

	Region	AVG_UNIT_PRICE	AVG_UNIT_COST
0	Asia	335.81	239.59
1	Australia and Oceania	222.67	154.74
2	Central America and the Caribbean	243.17	157.82
3	Europe	328.98	223.17
4	Middle East and North Africa	241.51	152.45
5	North America	277.24	205.29
6	Sub-Saharan Africa	259.62	183.68

In []: *#with chart*

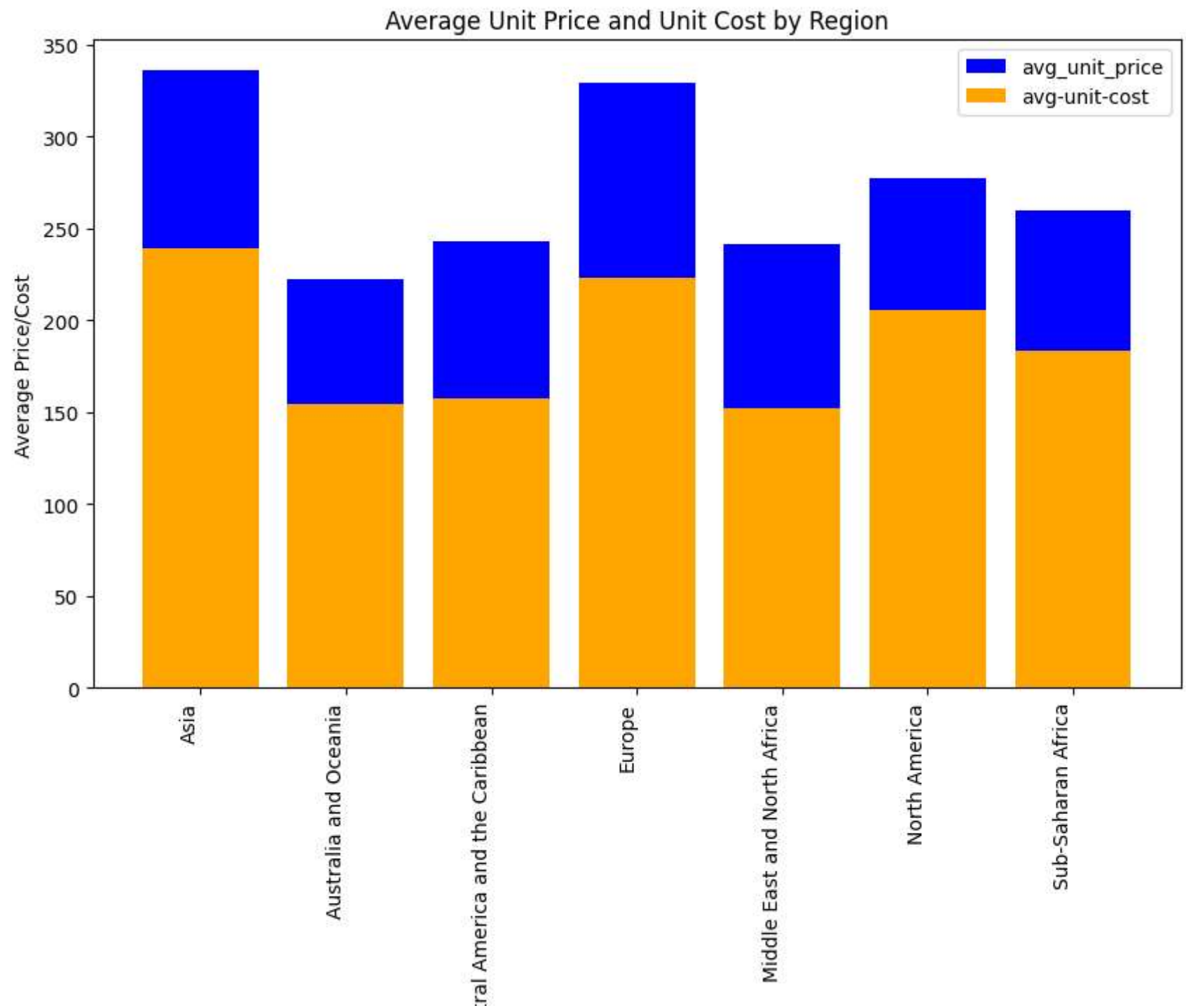
```

In [217... plt.figure(figsize = (10,6))

plt.bar(avg_price_and_cost_of_item['Region'], avg_price_and_cost_of_item['AVG_UNIT_PRICE'],
        label = 'avg_unit_price', color = 'blue')
plt.bar(avg_price_and_cost_of_item['Region'], avg_price_and_cost_of_item['AVG_UNIT_COST'],
        label = 'avg-unit-cost', color = 'orange')

plt.xlabel('Region')
plt.ylabel('Average Price/Cost')
plt.title('Average Unit Price and Unit Cost by Region')
plt.xticks(rotation=90, ha='right')
plt.legend()
plt.show()

```



Cent

Region

In []: 4. Which item type has the highest sales volume (units sold) overall?

```
In [150... query4 = """
SELECT [Item_Type] FROM [Amazon_sales]
Where [Units_Sold] = (SELECT MAX([Units_Sold])
FROM [Amazon_sales])"""

highest_sales_volume = pd.read_sql_query(query4, conn)
highest_sales_volume
```

Out[150... **Item_Type**

0	Baby Food
---	-----------

In []: 5. How many orders were placed in each sales channel, and which channel had the most orders?

```
In [164... query5 = """
SELECT [Sales_Channel], Count([Sales_Channel]) as Total_Orders FROM [Amazon_Sales]
GROUP BY [Sales_Channel]
ORDER BY [Total_Orders] DESC
"""

order_placed = pd.read_sql_query(query5, conn)
order_placed
```

Out[164... **Sales_Channel Total_Orders**

0	Offline	50
1	Online	50

In []: #with chart

```
In [199... plt.figure(figsize = (10,5))
plt.bar(order_placed['Sales_Channel'], order_placed['Total_Orders'])
```



```
plt.xlabel('Sales Channel')
plt.ylabel('Total_Orders')
plt.title('Total Orders By Sales Channel')
plt.show()
```



In []: 6. What is the distribution of order priority across all regions?

```
In [176... query6 = """SELECT [Region], [Order_Priority], COUNT(*) as Priority_Count FROM [Amazon_Sales]
            GROUP BY [Region], [Order_Priority]"""

order_priority_across_regions = pd.read_sql_query(query6, conn)
order_priority_across_regions
```

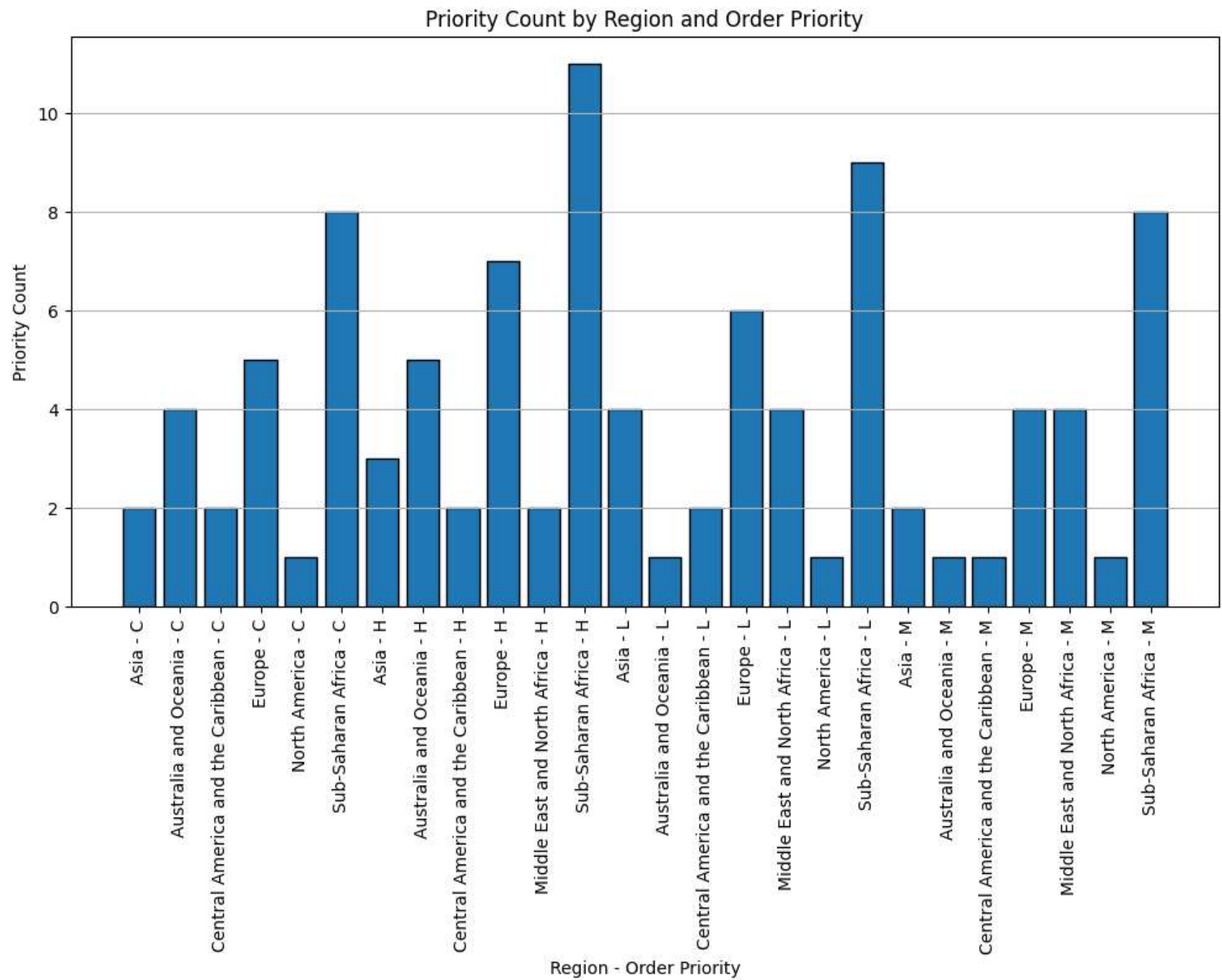
Out[176...

	Region	Order_Priority	Priority_Count
0	Asia	C	2
1	Australia and Oceania	C	4
2	Central America and the Caribbean	C	2
3	Europe	C	5
4	North America	C	1
5	Sub-Saharan Africa	C	8
6	Asia	H	3
7	Australia and Oceania	H	5
8	Central America and the Caribbean	H	2
9	Europe	H	7
10	Middle East and North Africa	H	2
11	Sub-Saharan Africa	H	11
12	Asia	L	4
13	Australia and Oceania	L	1
14	Central America and the Caribbean	L	2
15	Europe	L	6
16	Middle East and North Africa	L	4
17	North America	L	1
18	Sub-Saharan Africa	L	9
19	Asia	M	2
20	Australia and Oceania	M	1
21	Central America and the Caribbean	M	1

	Region	Order_Priority	Priority_Count
22	Europe	M	4
23	Middle East and North Africa	M	4
24	North America	M	1
25	Sub-Saharan Africa	M	8

In []: *#with chart*

```
In [183... plt.figure(figsize=(12, 6))
plt.bar(order_priority_across_regions['Region'] + ' - ' + order_priority_across_regions['Order_Priority'],
        order_priority_across_regions['Priority_Count'],edgecolor = 'black')
plt.xlabel('Region - Order Priority')
plt.ylabel('Priority Count')
plt.title('Priority Count by Region and Order Priority')
plt.xticks(rotation=90)
plt.grid(axis = 'y')
plt.show()
```



In []: 7. What is the average shipping time (in days) for orders from each country?

```
In [187... query7 = """SELECT [Country], AVG(DATEDIFF(Day, [Ship_Date], [Order_Date])) as Avg_Shipping_Time FROM [Amazon_Sales]
GROUP BY [Country]"""

avg_shipping_time = pd.read_sql_query(query7, conn)

avg_shipping_time
```

Out[187... **Country Avg_Shipping_Time**

0	Albania	-44
1	Angola	-4
2	Australia	-18
3	Austria	-7
4	Azerbaijan	-30
...
71	The Gambia	-17
72	Turkmenistan	-24
73	Tuvalu	-30
74	United Kingdom	-40
75	Zambia	-1

76 rows × 2 columns

In [188... 8. How does the average total profit vary across different order priorities?

Object `priorities` not found.

```
In [189... query8 = """
SELECT [Order_Priority], Round(AVG([Total_Profit]),2) as Avg_Total_Profit FROM [Amazon_Sales]
GROUP BY [Order_Priority]
"""
```

```
avg_profit_across_each_order_priority = pd.read_sql_query(query8, conn)

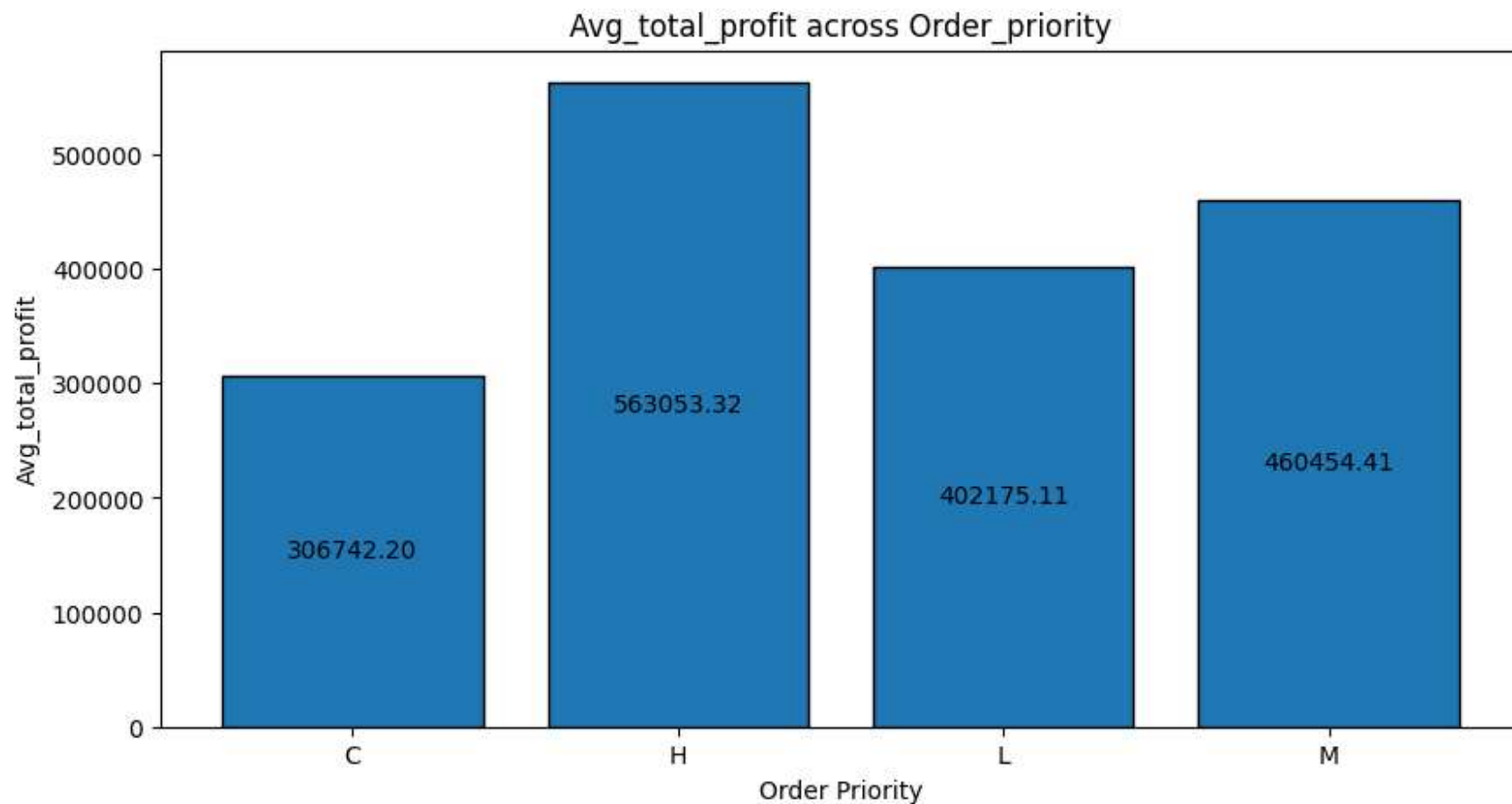
avg_profit_across_each_order_priority
```

Out[189... **Order_Priority** **Avg_Total_Profit**

	Order_Priority	Avg_Total_Profit
0	C	306742.20
1	H	563053.32
2	L	402175.11
3	M	460454.41

In []: *#with chart*

```
In [190... plt.figure(figsize = (10,5))
bar8 = plt.bar(avg_profit_across_each_order_priority['Order_Priority'],
               avg_profit_across_each_order_priority['Avg_Total_Profit'], edgecolor = 'black')
plt.bar_label(bar8, label_type='center', fmt='%.2f')
plt.xlabel('Order Priority')
plt.ylabel('Avg_total_profit')
plt.title('Avg_total_profit across Order_priority')
plt.show()
```



In []: 9. Which month had the highest total revenue, and which had the lowest?

```
In [196... query9 = """
SELECT MONTH([ORDER_DATE]) as Month , [Total_Revenue] FROM [Amazon_Sales]
WHERE [Total_Revenue] = (SELECT MAX([Total_Revenue]) FROM [Amazon_Sales])
or
[Total_Revenue] = (SELECT MIN([Total_Revenue]) FROM [Amazon_Sales])"""

highest_and_lowest_revenue_of_month = pd.read_sql_query(query9, conn)

highest_and_lowest_revenue_of_month
```

Out[196...

	Month	Total_Revenue
0	2	5.997055e+06
1	4	4.870260e+03

In []: *#with chart*

In [198...

```
plt.figure(figsize = (10,8))
bar9 = plt.bar(highest_and_lowest_revenue_of_month['Month'],
               highest_and_lowest_revenue_of_month['Total_Revenue'], edgecolor = 'black')
plt.yscale('log')
plt.bar_label(bar9, label_type = 'center', fmt='%.2f')
plt.xlabel('Month')
plt.ylabel('Total Revenue')
plt.title('Highest and Lowest Revenue Generated Month')
plt.show()
```